The Report Committee for Weiwei Ding

Certifies that this is the approved version of the following report:

Weakly Supervised Part-of-Speech Tagging for Chinese using Label Propagation

APPROVED BY

SUPERVISING COMMITTEE:

Supervisor: _____
Jason Baldridge

_____
Katrin Erk

Weakly Supervised Part-of-Speech Tagging for Chinese using Label Propagation

by

Weiwei Ding, B.A.; M.S.

REPORT

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF ARTS

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2011

Dedicated to my wife Meiying.

Weakly Supervised Part-of-Speech Tagging for Chinese using Label Propagation

Weiwei Ding, M.A.
The University of Texas at Austin, 2011

Supervisor: Jason Baldridge

Part-of-speech (POS) tagging is one of the most fundamental and crucial tasks in Natural Language Processing. Chinese POS tagging is challenging because it also involves word segmentation. In this report, research will be focused on how to improve unsupervised Part-of-Speech (POS) tagging using Hidden Markov Models and the Expectation Maximization parameter estimation approach (EM-HMM). The traditional EM-HMM system uses a dictionary, which is used to constrain possible tag sequences and initialize the model parameters. This is a very crude initialization: the emission parameters are set uniformly in accordance with the tag dictionary. To improve this, word alignments can be used. Word alignments are the word-level translation correspondent pairs generated from parallel text between two languages. In this report, Chinese-English word alignment is used. The performance is expected to be better, as these two tasks are complementary to each other. The dictionary provides information on word types, while word alignment provides

information on word tokens. However, it is found to be of limited benefit.

In this report, another method is proposed. To improve the dictionary coverage and get better POS distribution, Modified Adsorption, a label propagation algorithm is used. We construct a graph connecting word tokens to feature types (such as word unigrams and bigrams) and connecting those tokens to information from knowledge sources, such as a small tag dictionary, Wiktionary, and word alignments. The core idea is to use a small amount of supervision, in the form of a tag dictionary and acquire POS distributions for each word (both known and unknown) and provide this as an improved initialization for EM learning for HMM. We find this strategy to work very well, especially when we have a small tag dictionary. Label propagation provides a better initialization for the EM-HMM method, because it greatly increases the coverage of the dictionary. In addition, label propagation is quite flexible to incorporate many kinds of knowledge. However, results also show that some resources, such as the word alignments, are not easily exploited with label propagation.

Table of Contents

Abstract **v**

List of Tables **ix**

List of Figures **x**

Chapter 1. Introduction **1**

1.1 Part-of-Speech tagging 1

1.2 Chinese Part-of-Speech tagging 2

1.3 Incorporating more knowledge 4

Chapter 2. Data **7**

2.1 Chinese Penn Treebank 7

2.2 ISI Chinese English Parallel Text 8

2.3 Wiktionary 10

Chapter 3. Hidden Markov Model and Expectation Maximization **11**

3.1 Hidden Markov Model 11

3.2 Expectation Maximization 13

Chapter 4. EM-HMM based weakly supervised POS tagging **16**

4.1 Weakly supervised POS tagging with a dictionary 16

4.2 Improving POS tagging with word alignment 17

4.2.1 Mapping POS tags from English to Chinese 17

4.2.2 Combined Model 19

Chapter 5. Improving EM-HMM Using Label Propagation **24**
5.1 What is a better model 24
5.2 Using label propagation 25
5.3 Token-type model 30
5.3.1 Incorporating context information 30
5.3.2 Incorporating more language resources 31
5.3.3 Determining the weights for the links 33
5.3.4 The complete Model 35
5.3.5 Combined model LP+EM-HMM 36

Chapter 6. Tools and Experiment Settings **38**
6.1 Tools 38
6.1.1 Giza++ 38
6.1.2 Junto – the label propagation toolkit 39
6.2 Experiment Settings 40

Chapter 7. Results and Analysis **42**
7.1 System performance 42
7.1.1 EM-HMM system 42
7.1.2 Label propagation + EM-HMM system 43
7.2 Data analysis 47
7.2.1 In-domain vs out-of-domain 47
7.2.2 More analysis on results of label propagation 48
7.2.3 More discussions on Chinese POS tagging 49

Chapter 8. Conclusions and Future Research **51**
8.1 Conclusions 51
8.2 Future research 52

Bibliography **54**

List of Tables

2.1 Data splitting for CTB 8
2.2 Data splitting for ISI 9

4.1 A simplified example of the POS distribution for word tokens after combining word alignment 22

7.1 In-domain tests on CTB development set 42
7.2 Out-of-domain tests on ISI development set 43
7.3 Out-of-domain test on ISI development set using label propagation 44
7.4 In-domain test on CTB development set using label propagation 46
7.5 Out-of-domain test on ISI test set 46
7.6 In-domain test on CTB test set 47

List of Figures

1.1 Different word segmentation and POS tagging results on sentence "Water on the ground during raining days." 3

3.1 Hidden Markov Model 12

5.1 An example of the Token-Type model incorporating bigrams . 31
5.2 An example of the Token-Type model incorporating PFBigrams 32
5.3 An example of the Token-Type model incorporating Wiktionary 33
5.4 An example of the Token-Type model incorporating word alignment 34
5.5 The Token-Type model for label propagation 37

6.1 Comparison between gold alignments and Giza++ output . . 39

Chapter 1

Introduction

1.1 Part-of-Speech tagging

Part-of-speech (POS) tagging is one of the most fundamental and crucial tasks in Natural Language Processing. At first, this task may not seem so hard, because using simple models can achieve very high performance. Many models and technologies have been used for POS tagging. Based on whether training data is used, they can be classified into two categories. For unsupervised learning, Goldwater and Griffiths[10] used a Bayesian approach, and Gao and Johnson [7] worked on the Hidden Markov Model (HMM). For supervised learning, the maximum entropy model [28], support vector machine [8], and conditional random fields [20] are very popular.

In this report, weakly POS tagging based on HMMs with the Expectation Maximization (EM) parameter estimation method is in focus. The reason why this is chosen is because it is widely used and it is very convenient for us to add information into the system by simply providing different initial models to the EM method. The EM method only provides locally optimized results, so the initial model is very crucial. The simplest solution to provide EM an initial model is: extracting all possible POS tags from a dictionary for every word, and assigning uniform emission rates for each POS. However, this

solution does not generate expected results, especially for POS tagging, where the distribution for different tags of the same word are very imbalanced [13]. So the initial model for the EM method should be optimized.

Ravi and Knight [22] and Ravi et al. [23] both worked on how to optimize the initial model. The problem with the traditional use of dictionary is that many rarely occurring POS tags will have far more occurrences than in reality. So it is natural to think about how to remove these POS tags. The two papers used different strategies: the first used integer programming, and the second used greedy search. But the ideas are the same: trying to find out the tag bigrams which occur the most, because only the most frequent patterns in the data should be chosen. The low-frequency POS tags rarely occur, so they can be filtered out.

1.2 Chinese Part-of-Speech tagging

For Chinese, POS tagging is challenging, because it always involves another task, i.e. word segmentation. Figure 1.1 is an example from Jiang et al. [12]. The translation is added by the author. From this example, how word segmentation and POS tagging relate with each other can be seen. Chinese POS tagging is often based on the results of word segmentation. In this report, word segmentation is not in focus, but some experiments are based on the results of automatic word segmentation tools. So the results may be a little different from other research. Normally however, the performance of word

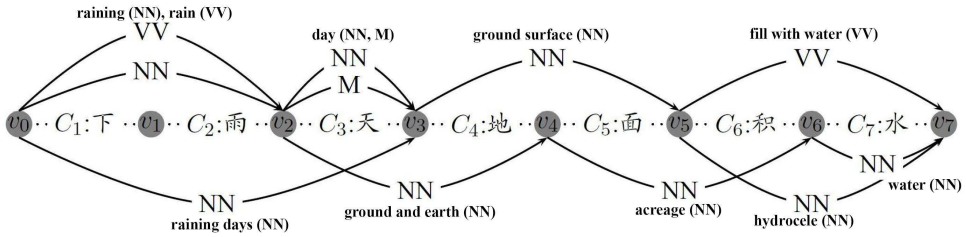

Figure 1.1: Different word segmentation and POS tagging results on sentence "Water on the ground during raining days."

segmentation tool is above 95% percent, so the influence of word segmentation is limited.

For Chinese POS tagging, most work has been done on supervised tagging. For example, Ng and Low [18] discussed whether it is better to solve word segmentation and POS tagging together than separately. Huang et al. [11] did research on using latent annotation and self-training to improve the performance of POS tagging. Besides these papers [11, 18], the research on unsupervised or weakly supervised Chinese POS tagging is still sparse.

The research by Cheng et al. [5] is a start on unsupervised POS tagging for Chinese. They tried to transplant some existed methods on POS tagging to Chinese. Their work was based on the HMM model, and three unsupervised parameter estimation methods were selected for parameter estimation: Expectation Maximization, Variational Bayes and Gibbs Sampling. Then tests were conducted to compared the performance of the three methods on Chinese unsupervised tagging. Tests showed that the performance on unsupervised Chinese POS tagging was very low, just around 24%. When they used a small POS tag set, the performance was around 48%. And tests showed

that the performance on Chinese was lower than English when using EM. They did not do much work to create a better model for the general unsupervised POS tagging problem or to incorporate more language-specific characteristics into the unsupervised methods. However, it is still a meaningful start on unsupervised POS tagging for Chinese.

1.3 Incorporating more knowledge

Besides using different models, another direction for improving unsupervised POS tagging would be providing more outside information. Previous research has shown that word alignment and POS tagging can benefit from each other. For example, Naseem et al. [24] tried to use word alignment results to improve POS tagging. Although they called their approaches "unsupervised," they took advantage of "supervised" results, as they utilized the full POS dictionary, containing a lot of POS knowledge. Their work revealed that the POS tagging task benefited from the word alignment task. What they did was to build a generative model, similar to the traditional Hidden Markov Model, except that above the POS layer, there was another alignment layer, which generated the POS. Toutanova [17] showed that the word alignment task could also benefit from the results of the POS task, but their approach was to incorporate the POS information into their supervised Hidden Markov Model based word alignment system.

The dictionary usage can be viewed as information about word types.

The dictionary tells something about the general principles of POS tags. The word alignment is information about word tokens. Because words with different POS tags would normally have different meaning, they often have different alignments. Hence word alignment can be viewed as information about word tokens, which reveals the specific use. If they can be joined together, the performance is expected to be better, because these two tasks are complementary to each other [17, 24].

There are other very useful language resources, such as Wiktionary, which is described in section 2.3. Wiktionary is a collaboratively created dictionary. It contains POS information. However, due to the huge differences between its labeling standards with that widely used in linguistic research, Wiktionary should be viewed more like a clustering result, than a linguistic dictionary. We thus must take care when incorporating such information into the systems.

These kinds of knowledge are very useful for the unsupervised tagging problem. Hence, a model which can easily incorporate all kinds of information would be useful. A good model about the task should be flexible enough to easily incorporate diverse knowledge, and benefit from it.

In this report, the label propagation algorithm Modified Adsorption [26] is used, in combination with a graph construction algorithm that connects word tokens to one another via intermediary words that act as features. These feature nodes serve two functions: a) they couple word tokens and put pressure on those tokens to have similar POS label distribution, and b) they couple to-

kens to declarative information from knowledge sources, such as Wiktionary.

Results show that using label propagation results in better performance than traditional HMM with EM estimation, especially when the POS tag dictionary is small. Label propagation provides a better initialization for EM because it greatly increases the coverage of the dictionary and generates better model parameters. With the incorporation of all kinds of knowledge, the performance is even better.

Chapter 2

Data

2.1 Chinese Penn Treebank

The POS dictionary is obtained from a corpus with POS tagging. The Chinese Penn Treebank is used as an annotated resource. The Chinese Penn Treebank provides 18783 sentences with full syntactic parses. In this report, only the POS tagging information is used. In the following experiments, "Chinese Penn Treebank" is simplified as "CTB".

Although the CTB is a high-quality corpus, labeling errors are still unavoidable. For example, the word "you" (have) has three POS tags: "VV", "VE" and "NN." However, "NN" only occurs once, and this is definitely a labeling mistake. Given that "you" is a high frequency word, and "NN" is a high frequency POS, many instances will be mistakenly labeled as "NN." This will lower the performance.

It is worth noting that CTB is not homogenous. It contains three newswire sources:

698 articles from Xinhua (1994-1998)

55 articles from Information Services Department of HKSAR (1997)

132 articles from Sinorama magazine, Taiwan (1996-1998 & 2000-2001)

All these sources are different on location and style. This will impact

Table 2.1: Data splitting for CTB

	CTB files	# of sentences
Training set	001-815, 1001-1136	64255
Development set	886-931, 1148-1151	802
Test set	816-885, 1137-1147	1903

the following tests.

In this report, the data splitting for CTB follows the setting used by Duan et al. [6], which is shown in table 2.1. They attempt to split the data from the three sources evenly into the training, development, and test set.

We seek to use as little manual input as possible, so the dictionary is not extracted from the whole set of training files, but only from a portion of it. Two settings are used: one is a dictionary extracted from the first 50 sentences in the training set, and the other is a dictionary extracted from the first 500 sentences in the training set.

2.2 ISI Chinese English Parallel Text

Another corpus we use is the "ISI Chinese-English automatically extracted parallel text" corpus ("ISI"). It is one of the Linguistic Data Consortium (LDC) resources, with catalog number LDC2007T09 and ISBN 1-58563-422-0. The data was extracted from news articles published by Xinhua News Agency and was obtained using the automatic parallel sentence identification

Table 2.2: Data splitting for ISI

	1-500	501-5000	5001-10000
Development set		√	
Test set	√		
Building tag mapping table			√

method. The corpus contains 558,567 sentence pairs. The sentences in the parallel corpus preserve the form and encoding of the texts in the original Gigaword corpora.

However, one shortcoming of the ISI data is that it only provides the aligned sentence pairs. There is neither information about Chinese word segmentation nor POS tagging information for both English and Chinese. For this research, this corpus has to be pre-processed to make it suitable for our task. The Standford Chinese word segmentor [29] and POS tagging tool [27, 28] were used to get the word segmentation and POS tagging results.

The ISI data contains much more data than needed, so only the first 10,000 sentence pairs are used for alignment. The data splitting for the 10,000 sentences is shown in table 2.2. The numbers in the first row indicates the id of the sentence pair in the 10,000 sentence pairs. The dictionaries used on the ISI data is the same as those on CTB.

2.3 Wiktionary

Wiktionary is an open dictionary project created and maintained by thousands of volunteers. Everyone can create, delete, and modify the dictionary entries. Specifically on Chinese, Mandarin Wiktionary contains more than 110,000 words. Every entry in Mandarin Wiktionary has the content as follows: the word, the Part-of-Speech information, English translation and some examples (optional). The following is an entry from the Mandarin Wiktionary:

Mandarin	Gaosu	Noun	# [[high]] [[speed]]
LanguageID	word	POS	English translation.

In this report, only the POS information is used. However, the POS tags in the Wiktionary are not the same with those in CTB. For example, there are 33 different POS tags in Wiktionary, while there are more than 40 in CTB, and the POS tagging standard of Wiktionary is quite different from that of CTB. So this cannot be directly used as our seed dictionary.

Chapter 3

Hidden Markov Model and Expectation Maximization

3.1 Hidden Markov Model

For POS tagging, the Hidden Markov Model (HMM) is one of the most widely used models. The HMM consists of two parts: the transition probabilities and the emission probabilities. In Figure 3.1 , the observed sequence (in this case, the observed sequence is the sequence of words) is represented by $y_1, y_2, y_3 \dots y_t$. The transition states (in this case, each state is the part-of-speech tag for each word) are hidden states, which are represented by $q_1, q_2, q_3 \dots q_t$.

The probability of the observed sequence $\vec{y}$ over the state sequence $\vec{q}$ is:

$$
\begin{aligned}
& P(\vec{y}, \vec{q}) = \\
& P(q1)P(y_1|q_1)(y_2|q_1q_2, y_1) \cdots (y_t|q_1q_2q_t, y_1y_2...y_{t-1}) \\
& P(q_2|q_1)P(q_3|q_1q_2) \cdots P(q_t|q_1q_2...q_{t-1}) \qquad (3.1)
\end{aligned}
$$

By Markov assumption, it is assumed that the current state is only impacted by the previous state, and the current observed label is only impacted

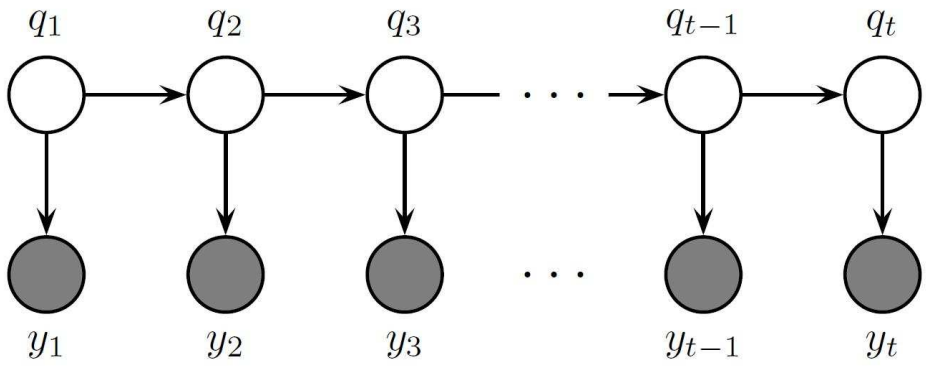

Figure 3.1: Hidden Markov Model

by the current state. So the formula can be changed to:

$$P(\vec{y}, \vec{q}) \approx P(q1) \prod_{i=1}^{t} P(y_i|q_i) P(q_{i+1}|q_i) \tag{3.2}$$

An HMM can be built from the labeled data by estimating the transition and emission rates. Using maximum likelihood estimation, the transition rate is estimated as follows:

$$P(q_i|q_j) = \frac{\sum \#(q_i, q_j)}{\sum \#(q_j)} \tag{3.3}$$

$\sum \#(q_i, q_j)$ represents the total number of occurrences for the tag bigram $q_i q_j$, and $\sum \#(q_j)$ represents the total number of occurrences for the tag unigram q_j. Using maximum likelihood estimation, the emission rate is estimated as follows:

$$P(y_i|q_j) = \frac{\sum \#(y_i, q_j)}{\sum \#(q_j)} \tag{3.4}$$

$\sum \#(y_i, q_j)$ represents the total number of occurrences when the observed label is y_i, and the state is q_j.

When the transition states are not known in the data, the parameters in the HMM are estimated using the expectation maximization (EM) method.

3.2 Expectation Maximization

Suppose we have an observed sequence Y and the set of possible states of the HMM, what we want is the model parameters: transition rates and emission rates. For this task, Expectation Maximization (EM) can be used to estimate the parameters for the HMM.

EM is a general algorithm for finding the maximum likelihood estimator of parameters in probabilistic models. It is an iterative algorithm, which contains two steps: E step and M step. In the E step, the expectation of the likelihood of the model is calculated based on the parameters of the old model. In the M step, the parameters that maximize the expected likelihood is calculated. With the two steps, the model is updated. It is worth pointing out that the likelihood function of the algorithm is not convex, so only a locally optimized model can be obtained.

Specifically for the HMM, the EM parameter estimation can be done through the forward-backward algorithm. In the following formulas, θ^n represents the old model we have at iteration n, θ^{n+1} represents the new model calculated through EM after iteration n. a_{ij}^n represents the transition rate from state i to state j, in model θ^n. b_{ik}^n represents the emission rate of observed label k given state q_i, in model θ^n. In the forward-backward algorithm, we also

need two supporting parameters: the forward probability α and the backward probability β. $\alpha_i^n(t)$ represents the forward probability at time step t being in state i under model n. $\beta_j^n(t+1)$ represents the backward probability at time step t+1 being state j under model n. The details about how to get α and β can be found in Manning and Schutze [15].

With all these parameters, the rate for transitional arc at time step t, from state i to state j, with observed label k, is calculated as follows (all parameters are calculated based on model θ^n, for simplicity, n is omitted. N represents the number of all possible states in the HMM, so $1 \leqslant$ i,j $\leqslant$ N):

$$\begin{aligned} p_t(i,j) =& \frac{\alpha_i(t) a_{ij} b_{ik} \beta_j(t+1)}{\sum_{m=1}^{N} \alpha_m(t)\beta_m(t)} \\ =& \frac{\alpha_i(t) a_{ij} b_{ik} \beta_j(t+1)}{\sum_{m=1}^{N}\sum_{g=1}^{N} \alpha_m(t) a_{mg} b_{mk} \beta_m(t+1)} \end{aligned} \tag{3.5}$$

After the probability of each arc is obtained, we can re-estimate the parameters for the HMM:

$$\begin{aligned} a_{ij}^{n+1} =& \frac{expected\ number\ of\ transitions\ from\ state\ i\ to\ j}{expected\ number\ of\ transitions\ from\ state\ i} \\ =& \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{t=1}^{T}\sum_{m=1}^{N} p_t(i,m)} \end{aligned} \tag{3.6}$$

$$\begin{aligned} b_{ik}^{n+1} =& \frac{expected\ number\ of\ emissions\ from\ state\ i\ to\ observed\ label\ k}{expected\ number\ of\ emissions\ from\ state\ i} \\ =& \frac{\sum_{\{t:y_t=k, 1\leqslant t\leqslant T\}}\sum_{m=1}^{N} p_t(i,m)}{\sum_{t=1}^{T}\sum_{m=1}^{N} p_t(i,m)} \end{aligned} \tag{3.7}$$

Then the new model is used as the starting model for the next iteration. By alternating between calculating the expectation of the log likelihood

of the model given the parameters and finding parameters that maximize the expected log likelihood, the model is updated, and the likelihood of the new model is greater than that of the old model.

Chapter 4

EM-HMM based weakly supervised POS tagging

4.1 Weakly supervised POS tagging with a dictionary

EM-HMM for POS tagging always requires an initial model. Normally, a dictionary is provided. The emission parameters of the initial model are set uniformly in accordance with the tag dictionary. Sometimes a complete POS dictionary is very hard to get and this makes this approach less useful. So instead of the complete POS dictionary, some researchers investigate performance with a small dictionary. For example, Ravi and Knight [22] used a dictionary with only the POS tags of the high-frequency words.

In this report, a dictionary is also used. The dictionary is obtained from CTB. The dictionary is obtained from the "training set" (for this approach, other statistical information from the training corpus is not used. It is only the source of dictionary).

If the word is not in the dictionary, all POS tags will be assigned to it. However, there are around 40 POS tags in CTB, but only some of the POS will be selected to build the default POS tag set. The standards are as follows:

1. If the POS tag represents a closed class, such as "DE" (for a special word de in Chinese), it will not be selected.

2. Rarely occurred tags, such as "FW" (foreign word), are not selected.

So the default POS tag set only contains the following tags: 'NN' (noun), 'NR' (proper noun), 'VV' (verb), 'CD' (number), 'NT' (organizations), 'JJ' (adjective). The reason why we want to filter out the other tags is because the coverage of the other tags is very low. In addition, when the size of the vocabulary is small, there are many out-of-vocabulary words. Ravi and Knight [22] showed that EM exploits a lot of rare tags and assigns them to common word types, and this lowers the system performance. To alleviate this effect, we have to filter out those low-frequency POS tags.

The dictionary approach is used as the start of our experiments. It is also called "EM-HMM" below.

Besides, the dictionary also provides us a way to generate the POS distribution for the words. A simple uniform POS distribution can be built for each word. The word is searched for in the dictionary, and for all the n possible POS tags it has, each will get a count of 1/n. For all the other POS tags, the count is 0. This simple POS distribution can be combined with the word alignment to further improve the performance of "EM-HMM."

4.2 Improving POS tagging with word alignment

4.2.1 Mapping POS tags from English to Chinese

Previous research showed that POS tagging and word alignment were complementary to each other [17, 24], so it is possible to improve POS tagging

using word alignment. In this paper, we choose Chinese-English alignment.

A study by Moon and Baldridge [16] showed that a Modern English corpus with POS information, along with an alignment model, could be used to label the Middle English corpus. They simply projected POS tags through alignments. Because the POS tag set of the Middle English corpus was different from the Penn Treebank tag set, they built a table of mappings from Middle English tags to Penn Treebank tags. For Middle English and Modern English, because they are different time variants of the same language, the difference is small. But Chinese and English are completely different languages. They even do not belong to the same language family. And the CTB tag set is quite different from the English Penn Treebank tag set. A simple table of mappings between the tag sets is not enough.

Besides, the sets of POS tags of Chinese and English are different, so some POS tags such as determiner in English do not exist in Chinese. To solve this, a probabilistic POS mapping table between the two languages is built. The probabilistic POS mapping table contains the conditional probability of a Chinese POS tag given an English POS tag when they are connected through alignments. For each Chinese word, when it is aligned with an English word, the POS tag of the English word will be searched for in the probabilistic POS mapping table, and what we get are the possible Chinese POS tags for the Chinese word, along with different probabilities.

Maximum likelihood estimation is used to generate a probabilistic POS mapping table. The mapping rate is calculated in the following way: N is the

number of possible POS tags in Chinese. CPOS represents the Chinese POS tag, and EPOS represents the English POS tag.

$$P(CPOS_i|EPOS_j) = \frac{\#of\ alignments(CPOS_i, EPOS_j)}{\sum_{m=1}^{N} \#of\ alignments(CPOS_m, EPOS_j)} \quad (4.1)$$

For 1-to-n alignments, each aligned POS pair will get a fractional count of 1/n. For example, for a 1-to-4 alignment, the count of each aligned pair is 0.25.

4.2.2 Combined Model

After the word alignment results are obtained, the problem is how the results are incorporated in the HMM and how to improve the performance.

In this report, linear interpolation is used to join the dictionary approach and the alignment results together. For each word token, a new POS distribution is calculated. For each Chinese POS, we can get the fractional count for the current word token from the dictionary: 1/n if the word token has this POS tag or 0 if it does not. Then for each English word the current token aligned with, we can get the count of this Chinese POS given the English POS tag of the aligned word, from the probabilistic POS mapping table built before. Linear interpolation is then used to join the two counts together. Then we normalize the result for each POS, and use it as the proportional count of each POS for the current word token. This is shown in the following formula.

$$P_{new}(CPOS_i|WT_k)$$

$$= \frac{\delta \sum_{j=1}^{J}(1/J * P_{align}(CPOS_i|EPOS_j)) + (1-\delta)P_{dict}(CPOS_i|WT_k)}{\sum_{i=1}^{N}(\delta \sum_{j=1}^{J}(1/J * P_{align}(CPOS_i|EPOS_j)) + (1-\delta)P_{dict}(CPOS_i|WT_k))} \tag{4.2}$$

where $0 \leqslant \theta \leqslant 1$

δ is used to adjust the weight of the word alignment information. EPOS_j represents the POS tag of each English word aligned with WT_k. J represents the number of English words aligned with WT_k. N represents all possible POS tags in Chinese. "WT_k" represents word token k. $\mathrm{P}_{dict}(\mathrm{CPOS}_i|\mathrm{WT}_k)$ represents the count of Chinese POS i given word token k generated from the dictionary, as discussed in the previous section. For all the n possible POS tags the word has, each will get a count of 1/n. For the POS tag the word does not have, the count P_{dict} is zero. $\mathrm{P}_{align}(\mathrm{CPOS}_i|\mathrm{EPOS}_j)$ can be searched from the probabilistic POS mapping table.

For example, if there are three POS: NN, VV, VE for the word token wt, the P_{dict} for each POS will be 1/3. Assume there is an alignment between this word and an English word, and that the English word only has one POS: NN. The NN tag can be mapped into three Chinese POS tags NN, NR, NT with rates 0.2, 0.3, 0.5. Suppose δ is 0.1. Then for this word token wt:

$$\begin{aligned} P_{new}(NN|wt) =& \frac{\delta \sum_{j=1}^{1}(1/1 * P_{align}(NN|NN)) + (1-\delta)P_{dict}(NN|wt)}{\sum_{i=1}^{N}(\delta \sum_{j=1}^{1}(1/1 * P_{align}(CPOS_i|NN)) + (1-\delta)P_{dict}(CPOS_i|wt))} \\ =& \frac{0.1 * 0.2 + 0.9 * 1/3}{\sum_{i=1}^{N}(\delta P_{align}(CPOS_i|NN)) + (1-\delta)P_{dict}(CPOS_i|wt))} \end{aligned} \tag{4.3}$$

If the word is not in the dictionary, then only the alignment information will be considered. If it is out-of-vocabulary, and it has no alignments, then the default POS tag set will be used and a uniform distribution is given.

After word alignment is used, we get the POS distribution for each word token. But for the HMM, what we need are the transition and emission rates. The calculation is presented in the formulas below. This strategy is also used for building an HMM from the results generated by label propagation. #(POS_{t_i}, POS_{t+i_j}) is the fractional number of the occurrence of tag bigram $\mathrm{POS}_i\mathrm{POS}_j$ for the word sequence $\mathrm{WordToken}_t$, $\mathrm{WordToken}_{t+1}$. #(POS_{t_i}) is the fractional number of tag unigram POS_i for $\mathrm{WordToken}_t$. p_{t_i} represents the probability of POS i for the word token t.

For example, the fractional number of bigram $\mathrm{POS}_1\mathrm{POS}_2$ for the sequence $\mathrm{WordToken}_3$, $\mathrm{WordToken}_4$ can be represented by #(POS_{3_1}, POS_{4_2}). It is p_{3_1}*p_{4_2}. Through the formulas below, the transition rates and emission rates of the HMM are estimated with maximum likelihood estimation based on these proportional counts. T represents the number of word tokens in text. The transition rate is estimated as follow:

$$\begin{aligned} P(POS_j|POS_i) =& \frac{\sum_{t=1}^{T} \#(POS_{t\_i}, POS_{t+1\_j})}{\sum_{t=1}^{T} \#(POS_{t\_i})} \\ =& \frac{\sum_{t=1}^{T} p_{t\_i} p_{t+1\_j}}{\sum_{t=1}^{T} p_{t\_i}} \end{aligned} \tag{4.4}$$

Table 4.1: A simplified example of the POS distribution for word tokens after combining word alignment

Token ID	Type	POS Distribution						
		NN	NR	VV	AD	PN	JJ	END
1	wo(me)	0.3	0.2	0	0	0.5	0	0
2	chi(eat)	0.3	0	0.6	0	0.1	0	0
3	ta(he)	0	0.2	0	0	0.7	0.1	0
4	bu(not)	0.1	0	0.1	0.8	0	0	0
5	chi(eat)	0.1	0	0.8	0	0	0.1	0
6	END	0	0	0	0	0	0	1.0

The emission rate estimation is estimated as follow:

$$\begin{aligned} P(WORD_i|POS_j) =& \frac{\sum_{\{t:word\ type\ for\ WordToken_t\ is\ WORD_i, 1 \leqslant t \leqslant T\}} \#(POS_{t\_j})}{\sum_{t=1}^{T} \#(POS_{t\_j})} \\ =& \frac{\sum_{\{t:word\ type\ for\ WordToken_t\ is\ WORD_i, 1 \leqslant t \leqslant T\}} p_{t\_j}}{\sum_{t=1}^{T} p_{t\_j}} \end{aligned} \tag{4.5}$$

A simplified example of the POS distribution after combining word alignment is shown in Table 4.1. An END tag is added for convenience. From this table, P(NN|NN) is calculated in the following way:

$$\begin{aligned} P(NN|NN) =& \frac{\sum_{t=1}^{5} \#(NN, NN)}{\sum_{t=1}^{5} \#(NN)} \\ =& \frac{0.3 * 0.3 + 0 * 0.3 + 0.1 * 0 + 0.1 * 0.1 + 0 * 0.1}{0.3 + 0.3 + 0.1 + 0.1} \\ =& 0.1/0.8 = 0.125 \end{aligned}$$

According to Formula 4.5, P(chi|VV) is calculated as follows:

$$\begin{aligned} P(chi|VV) =& \frac{\sum_{\{t:word\ type\ for\ WordToken_t\ is\ \text{“}chi\text{”},1\leqslant t\leqslant 5\}} \#(VV)}{\sum_{t=1}^{5} \#(VV)} \\ =& \frac{0.6+0.8}{0+0.6+0+0.1+0.8+0} \\ =& 1.4/1.5 = 0.933 \end{aligned}$$

This is very similar to building an HMM out of labeled data using maximum likelihood estimation. Building an HMM out of labeled data can be viewed as a particular case. In this case, the POS distribution for each word token is: 1 for only one POS tag and 0 for other POS tags.

After the model is built from the data containing POS distributions for each word, it is used as the new initial model for EM-HMM. We call this "EM-HMM+aligned".

Chapter 5

Improving EM-HMM Using Label Propagation

5.1 What is a better model

The key idea for this section is to build a better model initialization, such that the traditional EM-HMM model can be improved.

From previous discussions on the EM method, the likelihood function for EM is not convex, hence only a locally optimal solution can be obtained. So the initial model of the EM method is crucial. In the traditional EM-HMM method, the initial model is only determined by the dictionary, in which, all POS tags of each word are uniformly distributed. The drawbacks of the traditional EM-HMM method are obvious. First of all, the coverage of the dictionary is very limited. Because the dictionary is only extracted from very few sentences, many words are out-of-vocabulary. For these words, all POS tags are possible. Some strategies, such as removing the closed class tags from the candidate sets, can be used. However, nothing more can be done to solve this. Secondly, POS tags are treated uniformly in the traditional EM-HMM system. However, in fact, the POS tag distributions of words are highly imbalanced. The uniform distribution assumption is not an ideal start for EM. The incorporation of word alignments can alleviate such effects, because different

POS distributions can be given to word tokens, based on the aligned words they have. However, this is also not enough. Results show that the promotion for "EM-HMM + aligned" is still limited.

Correspondingly, a better initialization for the EM-HMM should fulfill the two following goals: broader word coverage and better POS distribution assumption. How to generate this out of the small dictionary is our main concern. This leads us to the idea of label propagation.

5.2 Using label propagation

Label propagation is one of the most widely used weakly supervised methods. It assumes a weighted graph, in which weights are non-negative numbers, indicating the importance of the linked nodes. The label propagation process repeatedly updates node tags by propagating labels from the neighbors. An example is shown below.

We can view all data points as nodes in the weighted graph. Nodes in labeled data and unlabeled data are connected to each other with different weights based on the similarity of each pair. w_{ij} is used to represent the weight of the link connecting node i and node j. Then the probability of node j propagating its label to node i is p_{ij}. p_{ij} is defined as follows:

$$p_{ij} = \frac{w_{ij}}{\sum_k w_{ik}} \tag{5.1}$$

k represents the number of neighbors node i has. We can see the greater the weight w_{ij} is and the lower the degree of node i is, the greater the prob-

ability p_{ij} will be. Let L be the set of possible labels, and $|L| = l$. g_j^t is a vector, which represents the output of label propagation for node j at time t. The value of each dimension g_{jl}^t represents how much we believe that the node j should have label l. For node i at time step t+1, summarizing all the neighbors yields the new possibility vector of node i:

$$g_i^{t+1} = \sum_k p_{ik} g_k^t \tag{5.2}$$

Label propagation is a general framework for propagating information from labeled data to unlabeled data. Based on this framework, many different algorithms are designed.

Label propagation has been proven to be very useful on many NLP tasks. Goldberg and Zhu [9] applied label propagation to address the sentiment analysis task of rating inference. Given a set of movie reviews and accompanying ratings (e.g., 4 stars), the task called for inferring numerical ratings for unlabeled reviews based on the perceived sentiment expressed by their text. Results showed that only when the labeled data was very limited, the label propagation achieved significantly better predictive accuracy over other methods that ignored the unlabeled examples during training. This proved that label propagation worked better with very sparse data by exploiting information from the unlabeled data. Baluja et al. [2] presented the Adsorption algorithm, and applied this algorithm to provide personalized video suggestions for YouTube users. Unlike previous label propagation algorithms, the Adsorption algorithm is a controlled random walk over the graph. Three different actions are used to formalize the control: inject, continue and abandon.

For each vertex v, there are three pre-defined probabilities for these three actions: $p_v^{inj}, p_v^{cont}, p_v^{abnd}$. Besides, $p_v^{inj}, p_v^{cont}, p_v^{abnd} \geqslant 0, p_v^{inj} + p_v^{cont} + p_v^{abnd} = 1$. p_v^{inj} represents the probability that the random walk stops and return the pre-defined information about the node. p_v^{abnd} represents the probability that this vertex is abandoned for the labeling process. p_v^{cont} represents the probability that the random-walk continues as normal. Then the node is updated summarizing the information provided by all the three possible actions. By using the Adsorption algorithm, the expected efficacy of suggestions in YouTube was improved. Talukdar and Crammer [26] proposed a new label propagation algorithm Modified Adsorption algorithm (MAD) based on the Adsorption algorithm. The biggest improvement of MAD is to construct an objective that reflects the three requirements of label propagation. The three requirements are:

- For the labeled vertices, we like the output of the algorithm to be close to the a-priori given labels.
- For pair of vertices that are close according to the input graph, we would like their labeling to be close.
- We want the output to be as uninformative as possible, this serves as additional regularization.

Then three hyper parameters μ_1, μ_2, μ_3 are used to balance between the three requirements. when $\mu_1 = 2 * \mu_2 = \mu_3 = 1$, MAD reduces to Adsorption algorithm. Test results on sentiment analysis and text classification showed that

this algorithm performed better than other label propagation algorithms. Experiments also showed that adjusting the three hyper parameters can improve system performance. This algorithm is what we use in this report.

Research by Subramanya et al. [25] was more closely related to this report. They applied a label propagation algorithm for semi-supervised training of conditional random fields (CRF) and applied it to POS tagging on a target domain, different from the domain where the training data was located. They used a similarity graph to encourage similar n-grams to have similar POS tags. Test Results showed that with access to the unlabeled target domain data, the label propagation based semi-supervised CRF performed better than the state-of-art supervised CRF. The results further prove that label propagation exploits information from the unlabeled data, which helps improve system performance.

The reason why label propagation is taken is quite natural. First, label propagation algorithms are very efficient on propagating tags from known data to unknown data [2, 9]. In this task, tags are different POS tags. The known data is the small dictionary. And the unknown data is the out-of-vocabulary words. It is known that all words can be connected through the contexts they appear. Through the contexts, POS tags can be propagated and the out-of-vocabulary words will have better POS distribution assumption than just an uniform distribution on all possible POS tags.

The second reason is about better POS distribution assumption, which is crucial in EM-HMM [22, 23]. The previous research on incorporating trans-

lation information [24] into EM-HMM inspired us with the idea on how to incorporate the word token information to get a better POS distribution. This can also be done in the propagation, because different word tokens have different contexts. And the tokens with the same POS tag should have similar contexts. For example, the word "hua" in Chinese has two meanings: one is flower (NN), and the other is spend (VV). For the tokens with POS NN, the contexts are: measure words, nouns related with plants, and some specific verbs such as "zhai" (pick). For the tokens with POS VV, the contexts are nouns related with money or time. Through these different contexts, word tokens with the two different POS tags can be differentiated. Taking advantage of the context information can provide a better POS distribution for different groups of word tokens.

The third reason is also very important. Label propagation algorithms are not just efficient, but also flexible. Many language resources can be incorporated into the label propagation system. For example, the Wiktionary POS can be used, because they can be the bridges connecting words of the same POS tags. The word alignment can also be used, because word tokens with the same POS should have similar translating correspondents. In the model described below, all these aspects of label propagation are exploited.

5.3 Token-type model

The idea of the token-type model is straightforward: the word dictionary is used as the seed information, i.e. the information about word types, and word tokens are the bridges connecting both seed information and different contexts and language resources. After label propagation is done, each word token is given a POS distribution. From the POS distributions given to word tokens generated by label propagation, an HMM initialization is built through the same strategy described in section 4.2.2. The formulas 4.4 and 4.5 are about how to estimate the transition rates and emission rates.

It is believed that this initial model will be better than the uniform POS distribution model in the traditional EM-HMM method, because it better describes the POS distribution and it covers more words.

5.3.1 Incorporating context information

The context information is easiest to access, and powerful in predicting the POS. Two types of context information are included.

N-gram is the most common technology used to describe the context information. To better balance between the data sparseness and power of description, bigram is used. The previous and the following bigrams are incorporated into this system. So for each word token, there will be two links connecting itself with its previous bigram and following bigram. An example is displayed in Figure 5.1.

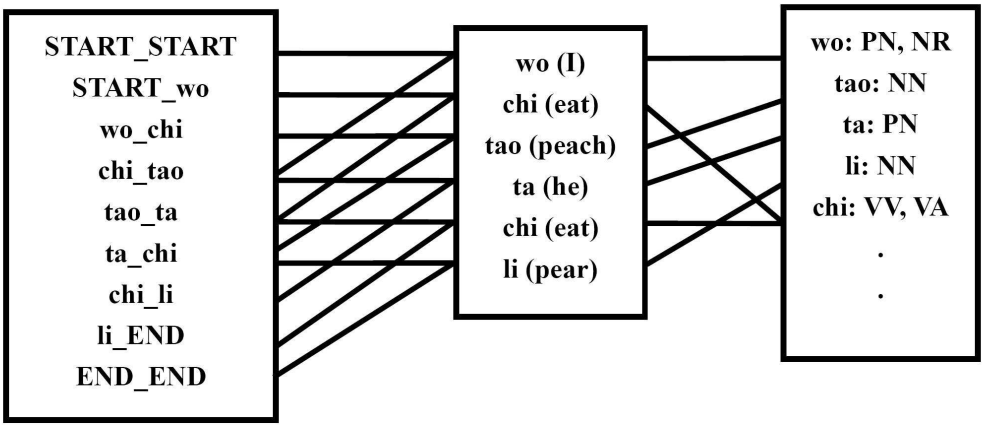

Figure 5.1: An example of the Token-Type model incorporating bigrams

Another variation of the bigram is the combination of the previous word and the following word of the target word. It is like the frame of the word. All words that fit in the same frame should share most in common on the POS. Tests in the next section show that this special bigram is very useful. For convenience, it is called "PFBigram". For each word token, there will be only one link connecting itself with its PFBigram. An example is shown in Figure 5.2.

To incorporate the context information, links will be generated between the words and the bigrams, PFBigrams.

5.3.2 Incorporating more language resources

Besides the contexts, two other resources are used for label propagation. One is Wiktionary, and the other is word alignment.

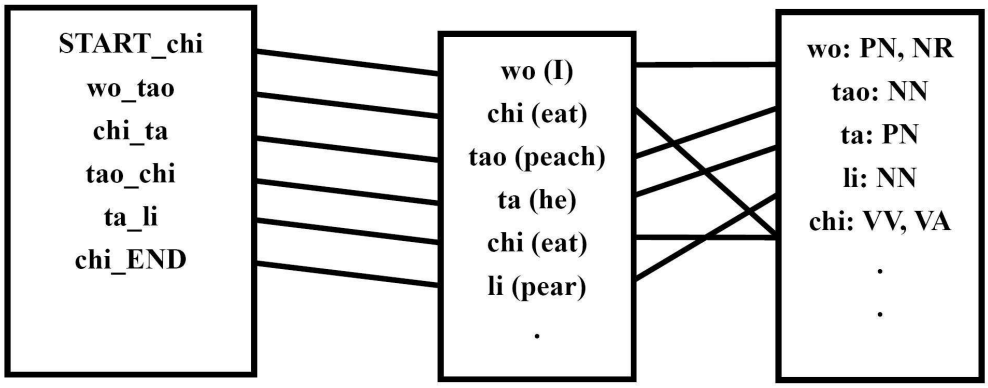

Figure 5.2: An example of the Token-Type model incorporating PFBigrams

Wiktionary is a open source dictionary. As discussed before, Wiktionary has different tagging standards for POS. But they can still be used as some information connecting words with similar POS. In label propagation, links are generated between the word tokens and each POS tag they have in Wiktionary. For each word token, because it may have more than one POS tag in Wiktionary, there may be more than one link connecting itself with each of its POS tags. An example is shown in Figure 5.3.

Word alignment is very powerful as described in the previous research, too. So in the label propagation method, it is also used. However, it is incorporated in a different manner. In the previous system, the information is used through building the probabilistic POS mapping table between Chinese POS tags and English POS tags. This is not ideal because we need 5000 fully POS tagged Chinese-English sentence pairs to build the probabilistic POS mapping table, and we also need POS information for the English sentences

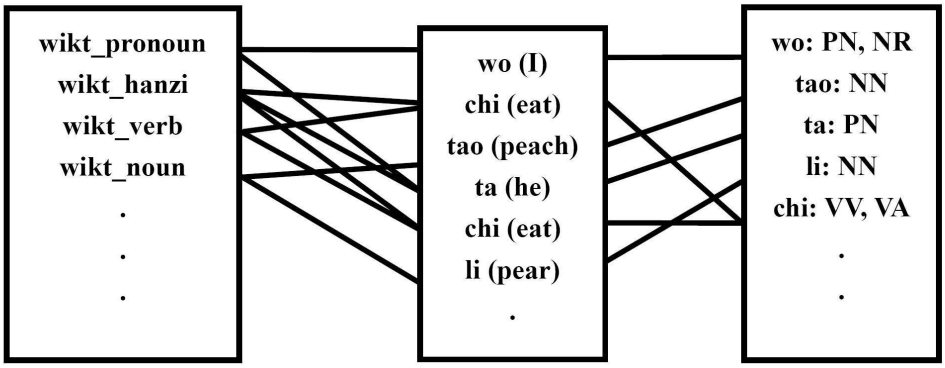

Figure 5.3: An example of the Token-Type model incorporating Wiktionary

during test. However, in the label propagation method used in this report, fully tagged Chinese-English sentence pairs are not needed, and English POS information is not needed, either. Links are generated between Chinese words and each English word they are aligned with. For each word token, because it may have more than one aligned word, there may be more than one links connecting itself with each word it is aligned with. An example is shown in Figure 5.4.

5.3.3 Determining the weights for the links

It is hard to determine how much weight to give to different links. The weights are the only parameter we adjust in our tests, so they are very crucial to the whole system, because they determine the portion of the information

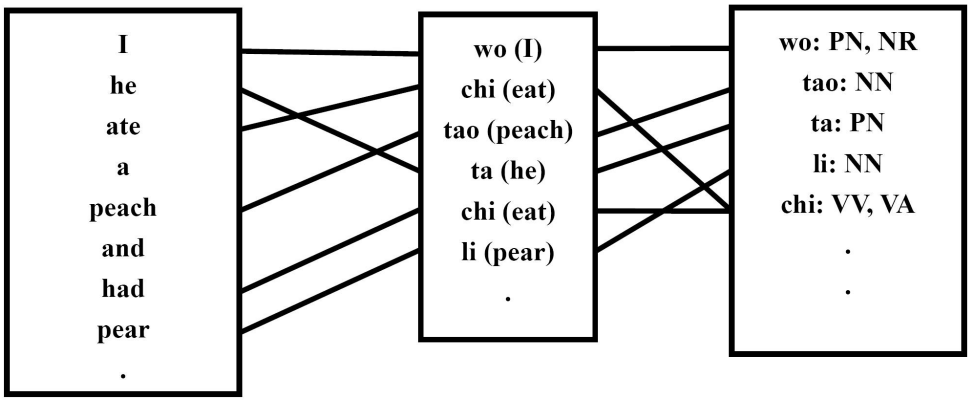

Figure 5.4: An example of the Token-Type model incorporating word alignment

that should be propagated.

We adjust the weight based on the type of the links. For all links between word tokens and word types, the weight is 1.0. For all links between word tokens and bigrams, PFBigrams, aligned words and Wiktionary POS tags, the weights are W_b, W_P, W_a, W_W respectively. Tests are conducted on the development set to adjust the weights. All the four weights are set between 0.2 and 2.0. Because the possible combinations of these weights are huge, we could not test all the possible combinations. Instead, we test each component separately, and get the best weights for each component respectively. Then we gradually add more components into the system, the order of the components being added is as below:

bigrams, PFBigrams, aligned words, Wiktionary POS tags.

Every time we set the weights of the existed component fixed, and gradually

change the weight of the new component from 0.2 to 2.0. When the highest performance is archived on the development set, this iteration is over. Then another new component is added into the system in the same way.

There are some one-to-many links between word tokens and other resources. For example, the token may have more than one POS tag in Wiktionary, and more than one aligned word. For these situations, all links connected with the same word tokens share the weight. Suppose that the word token is connected to 5 POS tags in Wiktionary and the weight for links connecting word tokens with the Wiktionary information is 1.0, then for every link between the word token and a POS tag, the weight is 0.2.

5.3.4 The complete Model

Based on all the previous discussions, the complete model used in this report is shown in Figure 5.5. In this figure, "Bigram" represents the previous and the following bigrams co-occurred with the target word. "PFBigram" represents the special bigram combining the previous word and the following word around the target word. "alignedW" represents the English words aligned with the target word. "wiktPOS" represents the POS tags of the target word in Wiktionary. "Type" represents the words in the dictionary. "Token" is each word token, the center node connecting all other nodes.

In this figure, besides the "Type" and the "Token" components, all other components are optional. In this research, all components will be added

one by one to test how well these components help the system performance.

5.3.5 Combined model LP+EM-HMM

The results of label propagation can also be used to generate the initial model for EM-HMM to further improve the system. After label propagation, each word token is given a POS tag distribution. Then the same strategy as in section 4.2.2 is used to build an initial model for EM-HMM.

However, the POS distribution generated by label propagation should not be used directly. One reason is that the POS distribution contains a special label "DUMMY". The "DUMMY" label [26] is designated explicitly to encode ignorance about the correct label. When the random-walk is abandoned, then the corresponding labeling vector is zero for all labels in the label set, and an arbitrary value of unit for the dummy label. The other reason is that many tags have very low rates. To solve these, two re-normalization strategies are designed. The first one will just extract the first 3 non-DUMMY tags, and then normalize their rates. The second one will extract all the non-DUMMY tags, if their rates are larger than 30% of the largest. Small experiments have been done, and the first one works better. In the following experiments, only the first strategy is used.

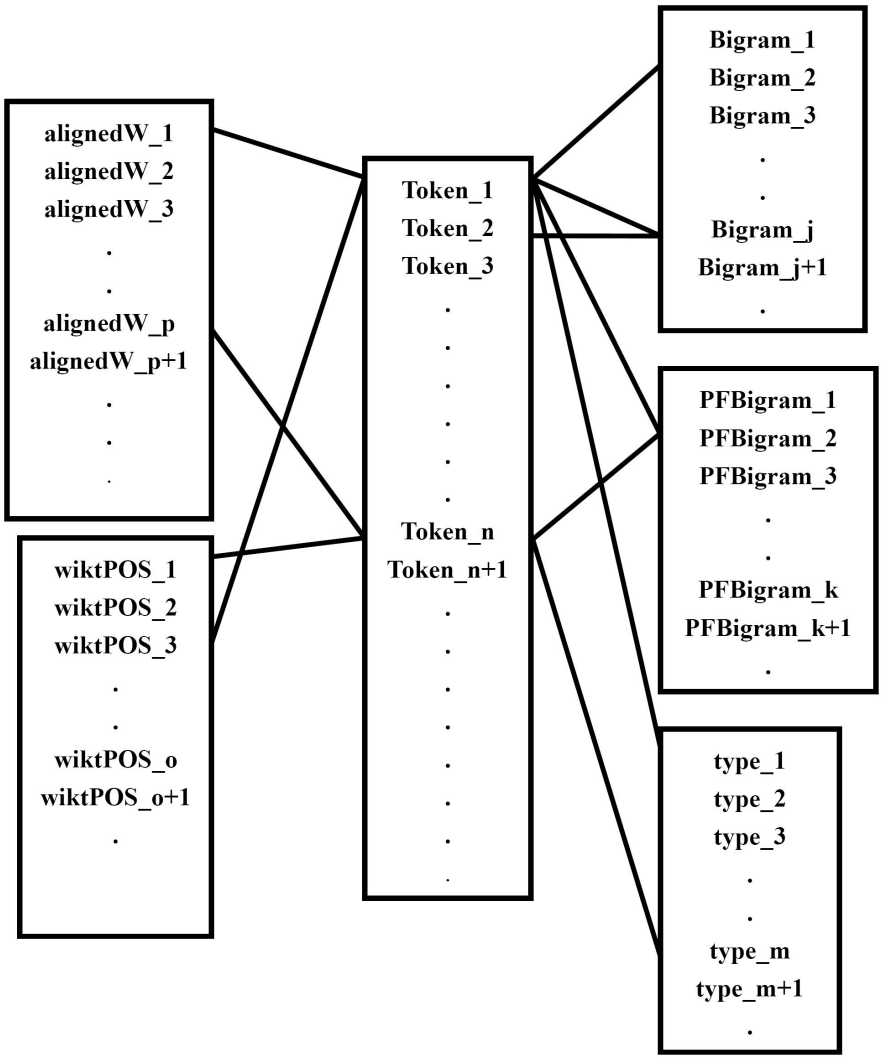

Figure 5.5: The Token-Type model for label propagation

Chapter 6

Tools and Experiment Settings

6.1 Tools

6.1.1 Giza++

Giza++ [19] is used to obtain the word alignments. To configure Giza++, the default parameter settings is used, which are a combination of IBM model 1, model 3 and model 4, and an HMM.

Besides 1-to-1 word alignments, Giza++ may generate 1-to-n alignments. Figure 6.1 shows one example of the Giza++ output. The alignments above are the gold alignments annotated by the author. The alignments below are the output of Giza++. From the figure, we can see the Giza++ output has two big problems. One is that many words in Chinese are not aligned. For all 12 aligned words in correct alignments, only 4 are aligned in the Giza++ output. The other is that for some words in Chinese sentences, Giza++ tends to give them much more aligned words than needed, such as "jiaoliang" (6 aligned words) and "pengbei" (4 aligned words) in the figure.

Although the output of Giza++ has very low recall, the precision is still not bad. So these alignments still provide much useful information. It is expected that if better alignments are provided, system performance will improve.

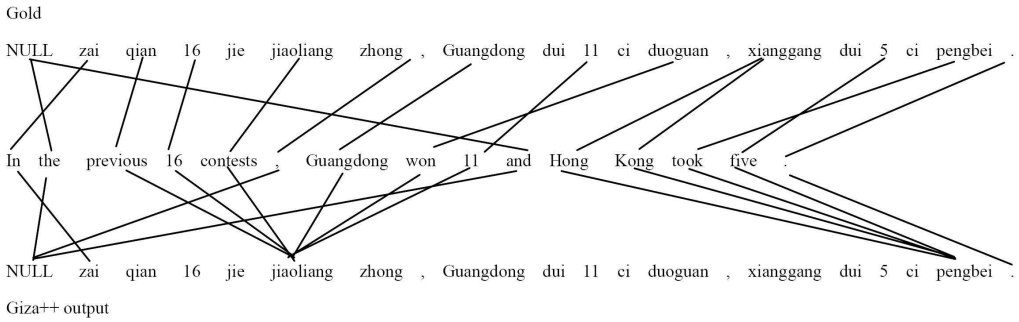

Figure 6.1: Comparison between gold alignments and Giza++ output

6.1.2 Junto – the label propagation toolkit

The Junto toolkit is implemented by Partha Talukdar. It consists of implementations of various graph-based semi-supervised learning (SSL) algorithms. Currently, three algorithms are implemented: Gaussian Random Fields (GRF) [30], Adsorption [2], and Modified Adsorption (MAD) [26].

In this report, Junto 1.0 is used, obtained from:

http://code.google.com/p/junto/

We choose the MAD algorithm and the iteration number for label propagation is 10.

6.2 Experiment Settings

In the following section, a series of experiments is reported on. To organize the experiments, three dimensions are used to describe the tests.

The tests are conducted on two kinds of data: one is on CTB, used as the in-domain test, and the other is on ISI, used as the out-of-domain test. These tests can show whether the methods are extensible on different data.

There are four approaches used in this paper:

- EM-HMM: the traditional EM-HMM system, only using a small dictionary.
- EM-HMM+aligned: the EM-HMM system incorporating alignment information
- LP: the system only using label propagation.
- LP+EM-HMM: the system combined label propagation with EM-HMM.

The first system "EM-HMM" is used as the baseline system. It incorporates the aligned information to improve the performance, to generate a different baseline system with higher performance, i.e. "EM-HMM+aligned", but this system can only work on the out-of-domain data with parallel English text. Label propagation can incorporate context information, Wiktionary information, and the alignment information. For the in-domain data, the aligned information is missing. Label propagation will generate the POS distribution for each word token. For comparison, the third system "LP" only takes

the non-DUMMY tag with the highest probability as the output label for each word token. "LP" can be viewed as another baseline system. The performance of "LP" indicates the efficacy of the label propagation approach. "LP+EM-HMM" uses the output of label propagation to build the initial model for the EM-HMM method. It is expected that "LP+EM-HMM" performs better than "LP" alone. Besides, in system "LP" and "LP+EM-HMM," the development set is also included during the process of label propagation on test set, because previous research [9, 25] showed that incorporation of unlabeled data generated better results. But for the following "EM-HMM" step, only the test set is used, and the development set data is unseen in the latter step. This is because for the "EM-HMM" step, we only want the labeling accuracy of the test set, and more data will make EM less efficient.

To see how much the dictionary can impact the results, two dictionaries are used in the tests: one is the dictionary extracted from 500 sentences, and the other is from 50 sentences. It is very likely that tests on the small dictionary perform worse than those on the large dictionary. But it is also expected that label propagation can alleviate the deficiencies of the small dictionary [9].

Chapter 7

Results and Analysis

7.1 System performance

In this section, the results of the four systems are displayed and discussed.

7.1.1 EM-HMM system

Table 7.1 shows the results of the in-domain test on the CTB development set. Two different dictionaries are used. "Dict-50" represents the dictionary extracted from the 50 sentences in the CTB training set, and "Dict-500" from 500. The system performs worse with "Dict-50" than with "Dict-500."

The results of the out-of-domain test on ISI development set are shown in Table 7.2. Compared to the results on the in-domain data, it is surprising that the performance on out-of-domain data is even better than that on

Table 7.1: In-domain tests on CTB development set

	Dict-50	Dict-500
EM-HMM	53.50%	64.83%

Table 7.2: Out-of-domain tests on ISI development set

	Dict -50	Dict-500
EM-HMM	67.01%	75.89%
EM-HMM + aligned	71.16%	78.34%

N.B: The labels being scored against in the ISI development set are produced by the Stanford tagger trained on the CTB data.

the in-domain data. More explanation will be given in the discussion section. Briefly speaking, it is because the CTB is not homogeneous, while the ISI is, and the labels in the ISI development set are produced by automatic POS tagger. Because the model used by the Stanford POS tagger is trained on CTB, the labeling style of automatic labeling results on the ISI data is very close to that of CTB. This leads to a better performance on ISI, compared to the previous test on CTB.

When word alignment is included into the system, the performance is improved. Both systems using "Dict-50" and "Dict-500" gain 3%-4% improvement.

7.1.2 Label propagation + EM-HMM system

With label propagation, it is expected that the performance will improve, because label propagation expands the coverage of the word dictionary, and better initial model can be built throughout the results of label propagation.

Table 7.3: Out-of-domain test on ISI development set using label propagation

	Dict-50		Dict-500	
	LP	LP+EM-HMM	LP	LP+EM-HMM
Bigram	57.54%	69.41%	65.96%	76.69%
PABigram	67.60%	70.77%	73.91%	78.12%
context	60.55%	69.77%	68.41%	78.30%
aligned	57.51%	62.04%	68.76%	73.07%
wikt	66.46%	72.18%	71.12%	78.78%
context+aligned	61.08%	70.80%	68.61%	78.39%
context+wikt	66.99%	73.77%	71.81%	79.08%
context+aligned+wikt	68.67%	**74.28%**	72.44%	**79.64%**

Table 7.3 shows the performance of out-of-domain tests on the ISI development set with different settings. "Context" means combining the "Bigram" and "PABigram" together. In Table 7.3, after using label propagation, the performance increases. Although the performance of "LP" is a little lower than that with EM-HMM, a very promising feed is built from the label propagation results for the following EM-HMM system to further improve the performance. Comparing the results with the traditional EM-HMM system, "LP+EM-HMM" produces a 4%-7% increase. The performance is slightly better compared with "EM-HMM+aligned". However, it is worth noting that the aligned information used in this system is not comparable to that used in "EM-HMM+aligned". This is because in "EM-HMM+aligned", the probabilistic POS mapping table is built on POS information, not just the word information. "EM-HMM+aligned" takes advantage of more POS information, making the comparison less fair.

Speaking about the in-domain data, results in Table 7.4 also shows that label propagation helps. Since there is no parallel text for the CTB data, aligned information is missing in this group of tests.

Compared to the tests on the out-of-domain data, label propagation performs better on in-domain data. The performance increase for "LP+EM-HMM" is 8%-13% on the development set. And "LP" alone works better than the baseline "EM-HMM" system, with a 7%-9% increase on performance. Explanation about this can be found in the following section. Besides, the output of "LP" is already so good that the following EM-HMM step does not improve much, only 1.3% increase when using small dictionary.

Specifically on each kind of information used in label propagation, the information from Wiktionary performs the best. Adding more information does not always result in an increase in performance. In the in-domain test with "Dict-500" shown by Table 7.4, "LP+EM-HMM" can generate the best results when using only Wiktionary, even better than using all kinds of information. This shows that only adjusting weights is hard to integrate all the information.

The results of the out-of-domain test on ISI test set are shown in Table 7.5. Unlike the development set, the tags in ISI test set are manually labeled, so the data in the test set has better quality.

From this table, we can see that when the dictionary is large, using "LP+EM-HMM" can generate a 6% increase on performance compared to "EM-HMM", and even when the word alignment is incorporated, "LP+EM-

Table 7.4: In-domain test on CTB development set using label propagation

	Dict-50		Dict-500	
	LP	LP+EM-HMM	LP	LP+EM-HMM
Bigram	56.68%	58.10%	64.59%	66.50%
PABigram	62.37%	59.31%	68.27%	65.15%
context	59.32%	59.73%	66.62%	66.88%
wikt	66.22%	64.67%	68.42%	**72.37%**
context+wikt	65.39%	**66.73%**	71.42%	71.60%

Table 7.5: Out-of-domain test on ISI test set

	Dict-50	Dict-500
EM-HMM	62.78%	71.77%
EM-HMM + aligned	64.16%	74.31%
LP	68.37%	71.64%
LP + EM-HMM	71.41%	77.66%

HMM" still outperforms "EM-HMM+aligned" by 3%. When the dictionary is small, label propagation performs even better. "LP" alone can beat "EM-HMM" and "EM-HMM + aligned" by 6% and 4% respectively. "LP+EMM" can even increase the performance by 3% more on the basis of "LP".

The results of the in-domain test on CTB test set are shown in Table 7.6. Because there is no parallel text for the CTB data, so results about "EM-HMM+aligned" are missing.

When using the small dictionary on the in-domain data, the largest performance gain is obtained. Compared to the "EM-HMM" system, "LP" increases the performance by 12%. The performance is even better, if followed

Table 7.6: In-domain test on CTB test set

	Dict-50	Dict-500
EM-HMM	53.48%	64.78%
LP	65.43%	69.16%
LP + EM-HMM	68.85%	72.43%

by EM-HMM. "LP+EM-HMM" outperforms "EM-HMM" by 15.5%. When using large dictionary "Dict-500", the performance gain is 4% and 7% respectively. This clearly demonstrates the power of label propagation on very little POS information.

7.2 Data analysis

7.2.1 In-domain vs out-of-domain

Explanation is needed to account for the performance difference on the in-domain and out-of-domain data. Normally, the performance on the in-domain data is higher. However, in this report, system performs better on the out-of-domain data. As mentioned before, CTB has 3 different sources: Xinhua news (mainland China), ISD (Hong kong) and Sinorama (Taiwan). The splitting of the training, development, test sets tries to balance between the three sources, which makes each set not homogeneous. But for the training set, only the first 50 or 500 sentences are used. All these sentences are from Xinhua News. For the ISI data, all sentences were extracted from Xinhua news. This is exactly the same source with the training set used in this paper.

So the performance on ISI data is better. In other words, the out-of-domain data is actually the "in-domain" data and the in-domain data "out-of-domain" because of the setting of the training set.

7.2.2 More analysis on results of label propagation

Label propagation is very efficient and powerful in predicting POS. Using label propagation alone can provide POS output with high performance. Besides, the results can be used to generate the initial model for EM-HMM. Experiments show that label propagation can help increase the performance of EM-HMM. When the dictionary is small, or the test set contains more raw data, or the test set is not homogenous, the label propagation is more powerful. Besides, label propagation is very flexible. All kinds of information can be easily incorporated into the system, to further improve the system.

However, the label propagation algorithm used in this report also has its own disadvantages. The most important one is that it does not perform so well on integrating language resources. This can be seen from the test results. For example, Wiktionary is a very useful source to provide solid information on POS. When more information is included, it does not always increase the system performance. This is because the only parameters adjusted in this report are the weights of the links; a fixed weight is given to all the links connecting the same type of nodes. To solve this, a better weight assigning strategy should be made. For example, for the high frequency words and the

low frequency words, the weight should reveal those differences. However, a better weight assigning strategy is hard to design.

7.2.3 More discussions on Chinese POS tagging

There are some problems specifically for Chinese, which makes the POS tagging harder than other languages. In this section, some of these problems are discussed below.

It is natural for many high-frequency Chinese words to have very low-frequency POS tags. One probability would be names. For example, the verb "mai" (buy) can be used as the surname. Similar examples include "zhang" (piece, open). Another possibility is that different words sometimes share the same character. For example, "hua" can represent the noun "flower" and the verb "spend." "Suo" can represent the verb "lock" and the noun "locker." Although this can be easily solved by some long distance context information, it is the disadvantage of the HMM, as because only takes advantage of the previous POS tag.

In Chinese, words have no boundaries. This makes Chinese POS tagging trickier than other languages, in which words have clear boundaries. Arguments have been made on whether it is better to do word segmentation and POS tagging together or separately [18]. In this report, POS tagging is conducted on the gold-standard segmentation. However, it is interesting to test whether doing the two tasks together would perform better.

The last problem is, that in Chinese, there are many word usages which actually come from ancient Chinese. The different time layers make POS tagging even harder.

Chapter 8

Conclusions and Future Research

8.1 Conclusions

In this report, discussions on ways to improve weakly supervised POS tagging using EM-HMM is provided. EM-HMM with a word dictionary is a widely used approach on weakly supervised POS tagging. However, because the word dictionary is limited in coverage, and does not make differentiation on the different POS tags one word has, the performance is generally poor. Besides, this model is not flexible enough to incorporate all kinds of information.

To solve these problems, label propagation is introduced. The context information, Wiktionary, and word alignment are used to connect different word tokens. With label propagation, it is expected that word tokens occurring in the same context, sharing the same POS information in Wiktionary, aligned with the same English words, are connected to each other, so that the POS information are propagated among them.

Results show that this approach increases the system performance substantially. This approach performs even better when the data is not homogeneous with the training set, and/or when the dictionary is very small. It is also shown that label propagation is so flexible that all kinds of information could

easily be incorporated. However, the label propagation method used in this report also has drawbacks. It performs poorly on integrating many resources. Part of the reason is that the weights of links are only determined by the sort of resource, and the characteristics of the word itself, such as frequency, are not considered.

From this research, we can see that the initial model for EM-HMM is the most crucial factor in building a better system. An initial model which contains more useful information, such as a larger dictionary, generates better results.

8.2 Future research

Results show that label propagation is very useful on weakly supervised POS tagging. In addition to being flexible, it has very promising future as the idea of label propagation fits well with the POS tagging task, and it is flexible. But to further improve the system performance, a better weight assigning strategy is needed. Some experiments have been done on using conditional probability as the weight of the context information, but the performance is not better compared to the strategy used in this report. Using a directed graph, i.e., giving different weights on different directions of propagation did not work either. Besides, the directed graph for label propagation increases the memory use. When the size of the data is small, for example, when the data contains less than 5000 sentences, the memory consumption is accept-

able. But more sentences are a great burden on the system resources when using a single processor. It remains challenging to build an efficient, effective-on-integration-of-information label propagation system.

Bibliography

[1] Vamshi Ambati, Stephan Vogel, and Jaime Carbonell. Active semi-supervised learning for improving word alignment. In *In Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, 2010.

[2] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, pages 895–904, New York, NY, USA, 2008. ACM.

[3] Thorsten Brants. Tnt - a statistical part-of-speech tagger. In *In Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP 2000)*, 2000.

[4] Eric Brill. Some advances in transformation-based part of speech tagging. In *National Conference on Artificial Intelligence*, pages 722–727, 1994.

[5] Alex Cheng, Fei Xia, and Jianfeng Gao. A comparison of unsupervised methods for part-of-speech tagging in chinese. In *Coling 2010: Posters*, pages 135–143, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[6] Xiangyu Duan, Jun Zhao, and Bo Xu. Probabilistic models for action-based chinese dependency parsing. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 559–566, Berlin, Heidelberg, 2007. Springer-Verlag.

[7] Jianfeng Gao and Mark Johnson. A comparison of Bayesian estimators for unsupervised Hidden Markov Model POS taggers. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 344–352, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[8] Jesus Gimenez and Lluis Marquez. Fast and accurate part-of-speech tagging: The svm approach revisited. In *Recent Advances in Natural Language Processing - RANLP*, pages 153–163, 2003.

[9] Andrew Goldberg and Xiaojin Zhu. Seeing stars when there arent many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 45–52, New York City, June 2006. Association for Computational Linguistics.

[10] Sharon Goldwater and Tom Griffiths. A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 744–751, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[11] Zhongqiang Huang, Vladimir Eidelman, and Mary Harper. Improving a simple bigram hmm part-of-speech tagger by latent annotation and self-training. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 213–216, Boulder, Colorado, June 2009. Association for Computational Linguistics.

[12] Wenbin Jiang, Haitao Mi, and Qun Liu. Word lattice reranking for chinese word segmentation and part-of-speech tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 385–392, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

[13] Mark Johnson. Why doesn't EM find good HMM POS-taggers? In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 296–305, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[14] Philipp Koehn. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand, 2005. AAMT, AAMT.

[15] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, MA, 1999.

[16] Taesun Moon and Jason Baldridge. Part-of-speech tagging for middle English through alignment and projection of parallel diachronic texts. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 390–399, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[17] Tahira Naseem, Benjamin Snyder, Jacob Eisenstein, and Regina Barzilay. Multilingual part-of-speech tagging: Two unsupervised approaches. *Journal of Artificial Intelligence Research*, 36:341–385, 2009.

[18] Hwee Tou Ng and Jin Kiat Low. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 277–284, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[19] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Comput. Linguist.*, 29:19–51, March 2003.

[20] Xuan-Hieu Phan. Crftagger: Crf english pos tagger. 2006.

[21] Adwait Ratnaparkhi. Maximum entropy models for natural language ambiguity resolution. Technical report, University of Pennsylvania, 1998.

[22] Sujith Ravi and Kevin Knight. Minimized models for unsupervised part-of-speech tagging. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference*

on Natural Language Processing of the AFNLP, pages 504–512, Suntec, Singapore, August 2009. Association for Computational Linguistics.

[23] Sujith Ravi, Ashish Vaswani, Kevin Knight, and David Chiang. Fast, greedy model minimization for unsupervised tagging. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 940–948, Beijing, China, August 2010. Coling 2010 Organizing Committee.

[24] Benjamin Snyder, Tahira Naseem, Jacob Eisenstein, and Regina Barzilay. Adding more languages improves unsupervised multilingual part-of-speech tagging: A bayesian non-parametric approach. In *In Proceedings of NAACL/HLT*, 2009.

[25] Amarnag Subramanya, Slav Petrov, and Fernando Pereira. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 167–176, Cambridge, MA, October 2010. Association for Computational Linguistics.

[26] Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *ECML/PKDD (2)'09*, pages 442–457, 2009.

[27] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-rich part-of-speech tagging with a cyclic dependency net-

work. In *North American Chapter of the Association for Computational Linguistics*, 2003.

[28] Kristina Toutanvoa and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 63–70, Hong Kong, China, October 2000. Association for Computational Linguistics.

[29] Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. A conditional random field word segmenter for sighan bakeoff 2005.

[30] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *IN ICML*, pages 912–919, 2003.